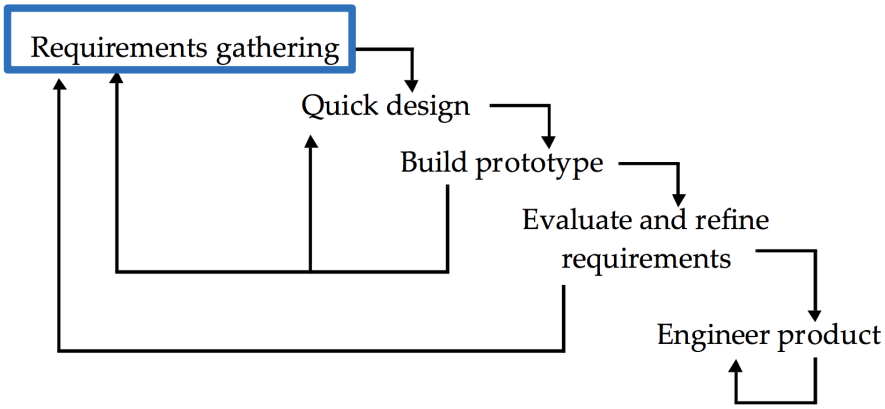
Part IA Interaction Design Notes

Iterative user-centred design and development



1 Requirement Gathering

- A requirement is a statement about what and how a future product should perform
- Collect sufficient, relevant, and appropriate data
- Two types of requirement
  1. Functional Requirements - What the system should do
  2. Non-functional Requirements - What are the constraints of the system - focus on user
     - Data Requirements
       - Where data is from, what type, how it is stored, and how long for
     - Environmental Requirements
       - Physical, social, organisational, and technical characteristics
     - User Characteristics
       - IT skills, age, gender, nationality, disabilities, and preferences
     - Usability Goals
       - Effectiveness, efficiency, safety, utility, etc.
     - Experience Goals
       - Enjoyable, etc.
- Two methods of requirement analysis
  1. Socio-technical methods
  2. Soft systems methodology (SSM)

1.1 Socio-technical Analysis

- Stateholds
  - Anyone who is effected by the success or failure of the system
- Considers social and technical issues side-by-side
- Made up of two models

1. CUSTOM Model - identify stakeholders
   - Primary - People who use the system
   - Secondary - People who produce input or receive output from the system
   - Tertiary - People who are effected by the success or failure of the system but aren't primary or secondary
   - Facilitating - People involved in the system's design, development, and maintenance
2. Requirements Development - Analysis the stakeholder's characteristics in terms of
   - Aims - What they want to achieve, and how measure it
   - Job Satisfaction - Sources of job satisfaction/ stress
   - Knowledge and Skills - What do they have
   - Work-group Attributes - What attributes the work-group have
   - Features of activity - How frequent, fragmented is the task
   - Responsibilities - Are there any considerations on responsibilities, security, or privacy
   - Work conditions - What are the conditions

1.2 Soft Systems Methodology

- Broader view, look at larger context
- Learn about problem situation rather than solve a pre-defined problem
- Three stages
  1. Rich Picture
     - A detailed description of the problem situation
     - Who are the stakeholders? What groups do they work in? What are their tasks?
     - Developed by interviews, observations, and workshops
  2. Root Definitions - Stakeholder perceptions
     - What stakeholders perceive to be the activities taking place
     - Can be several, for each stakeholder
     - CATWOE to list perceptions
       * Clients - People that benefit or accept output from the system
       * Actors - Stakeholder that perform activities in the system
       * Transformations - What changes the system performs on things
       * World-view - How the system is perceived by a client
       * Owner - Who owns the system/ can make changes
       * Environment - What factors influence the system
  3. Conceptual Model - What the system must do to meet the root definitions
     - Most important root definition is transformations, used in conceptual model
     - Define what is achieved and how
     - Conceptual model used to identify differences between the real world situation and how it is perceived

1.3 Data Gathering Methods

- The choice of a data gathering method depends on the amount of time available, level of detail required, and the level of risk
- Observations

  - Observations in the field

Pros	Cons
Realistic setting and activities	Difficult to setup
	Complex observation may intrude upon the user
	Problems of privacy and reliability

  - Observations in the lab

Pros	Cons
Less intrusion	But not realistic setting and activities

  - Direct observations - Users say what they are doing it while they do it

Pros	Cons
Good for understanding the nature and context of the task	too much data, time consuming

  - Indirect observations - Users explain what they did

- Interviews - An interviewer asks an interviewee set of questions, constrain interview length
  - Unstructured interview (open questions)

Pros	Cons
Rich data	Not too much data
	Off topic, long

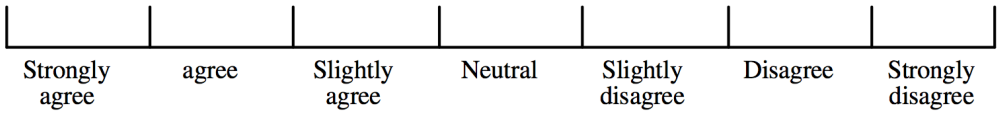
  - Structured interview (closed questions)

Pros	Cons
Easy data collection	Rigid

  - Semi-structured interview (open and closed questions)

Pros	Cons
Rich structured data	

- Focus groups - Interview people in groups
  - Make participants be representative of target users
- Card sorting
  - Participants given a pack of cards
  - Get them to put the cards into groups
  - Open card sorting - Given cards, with no pre-established groups
  - Closed card sorting - Given cards, with pre-established groups
  - Hybrid card sorting - Combination of the two
  - Number cards, and record these to record group
- Questionnaire
  - Designed to get specific information from a large and varied group of people
  - Open Questions - free to write their own answers
  - Closed Questions - select answer from a set of presented questions
  - Likert scale



  - Provide incentive, no longer than two sides of A4

- Studying documentation - procedures and rule often written down

Pros	Cons
Availability	Idealised reports
Accessibility	Outdated

- Scenario and user cases
  - Scenario - description of a flow of activity
  - Use case - scenarios focusing on intended interaction flow
- Researching similar products

Pros	Cons
Helps prompt requirements	May inhibit creativity
Helps generate alternative designs	

1.4 Guidelines

- Focus on identifying the stakeholder's needs
- Involve all stakeholder groups, with more than one representative from each group
- Use a combination of data gathering techniques
- Think about how to record the data
- Support this process with prototypes

1.5 Data Analysis and Interpretation

- Quantitative analysis - numerical methods to determine size, magnitude, amount, etc.
  - Averages - mean, median, and mode
  - Percentages
  - Graphical representation
  - For card sorting, either by eye or cluster analysis
- Qualitative analysis - Express the nature, themes of responses; try to quantise responses
  - Look for recurring patterns or themes
  - Look for key events
  - There are a number of different frameworks to analyse this
- Tools to support analysis include spreadsheets, statistical packages, and Qualitative data analysis tools

2 Quick Design

- Aims: to optimise the user's interaction with the system so it extends the user's activities in a useful, efficient, and usable way
- User centred design involves users participating in the design process
- Participatory design
  - Work Focused
    - Focuses on improving worker's environments
  - Collaborative
    - The designers and users collaborate on the design
  - Iterative
    - Several design and evaluation phases
- Participatory design uses models and techniques to communicate ideas
- Develop alternative designs that meet the user's needs
  - Design - Suggesting ideas for meeting the identified requirements
  - Conceptual Design - What the system will do - abstract
  - Physical Design - Details of the design
- Alternative designs will have the same conceptual design but different physical designs
- Participatory design techniques
  - Brainstorming
    - Everyone comes up with lots of ideas
  - Concept development
    - Find the driving concept or metaphor behind the design
  - Mood Board
    - A collage of pictures and text
  - Prototyping
    - Lets users interact with the design
    - Used to demo a concept, develop a design, user testing
    - Used to design workflows, technical aspects, screen layouts, look-and-feel, and critical areas (security)
    - Two types of prototype
      1. Lo-fi
         - Rough, made of paper or similar, easily iterated on
         - Can make a video of the system in use
         - Form model - Does not work but shows form
         - Wireframe - Schematic of information architecture
         - Working prototype - works, but with limited functionality, form may not be accurate
         - Wizard-of-Oz - Someone controls the system while it is in use
      2. Hi-fi - Well fleshed out prototype, not as good as Lo-fi
         - Users often think its final product
         - Focus on small details

  - Storyboarding
    - Idea of user's activities presented in a storyboard
  - Workshops
    - Forum for discussion
    - Designers and users can ask each other questions
    - Used to fill in gaps of understanding

2.1 Good Design

- Concerned with usability of interfaces
  - Usability - How easily users can learn and use a product to achieve their goals and how satisfied they are with it
  - A mixture of engineering and human user aspects
- Good UIs are
  - Easy to learn
  - Easy to remember
  - Predictive
  - Easy to recover from errors
  - Efficient
  - Engaging
- Design Principles - High level and context free design goals based on theorems
- Design Guidelines - Specific and context dependant rules for designers to follow to achieve the principles
- The principles of good design
  1. Recognise Diversity
     - Accommodate for all different types of user (old, young; iOS, Android; novice, expert)
  2. Follow the golden rules of interface design
     - Follow the eight principles listed below
  3. Prevent Errors
     - Either prevent users from performing errors or allow them to undo these actions
  4. Follow the Guidelines for Data Display
     - Consistency - terms, colours, formats, etc.
     - Efficient information assimilation - Data outlining, comprehensible labels
     - Minimal memory load - no need to remember information from screen to screen
     - Provision of user control of data display - font size, sorting of rows, contrast, etc.
     - Getting user attention - subtle and appropriate
  5. Follow the Guidelines for Data Entry
     - Consistency - similar style for similar data
     - Minimal actions - selection rather than freestyle typing, automatic completion
     - Minimal memory load
     - Provision of user control of data entry - experts may prefer command language
  6. Balance Automated and Human Control
     - Tedious and repetitive - give to robots

  - Decision making and creative - give to humans

- The Golden Rules of interface design
  1. Consistency - Of terms, icons and data / command flow
  2. Universal Usability - Scale use flow with user experience (keyboard shortcuts)
  3. Informative Feedback - For every user action, there should be some feedback from the system
  4. Dialogues with closure - Interactions should have a beginning, middle, and end
  5. Prevent Errors - Design so users can't make serious errors
  6. Reversal of Actions - Have an undo command
  7. User in Control - The user should have control at every point in the execution of an application
  8. Reduce short term memory - Don't make users remember too much information

2.2 Cognitive factors in design

2.2.1 The model human processor

- Made up of three sub-systems
  - Perceptual - Performs input and output
    - Input via senses, output via movement
    - Information is stored in memory
  - Cognitive - Information is processed and meaning applied
    - Performs reasoning, problem solving, skill acquisition, and error feedback
  - Motor - Produces reactions
    - Using motor skills as a result of cognitive processing
- Perceptual slower than cognitive and motor
- Allows a system designer to predict the time it takes a person to complete a task

2.2.2 Memory and Cognition

- Three types of memory, sensory buffers, long term, and working memory
- Attention moves information from the sensory buffers into working memory
- Can not expect people to remember everything they have seen
- Will stay in working memory for about 10 seconds
- Short term memory has a limited capacity of 7 ± 2 chunks of information
- Items at the beginning and end of a list are remembered best
- People store meaning or knowledge in long-term memory
- The amount remembered is proportional to the amount of time spend rehearsing
- Forgetting
  - Decay - information is lost slowly over time
  - Interference - new information replaces old
- Constructivism
  - Mind creates models instead of pixel-by-pixel images
  - Context and mindset play a large role in what people see and hear

2.2.3 Gestalt Theory

1. Figure-ground relationship - we group elements as either figures or background
2. Proximity - we group by distance or location
3. Similarity - we group by type
4. Symmetry - we group by meaning
5. Continuity - we group by flow
6. Closure - we perceive shapes that aren't completely there

2.2.4 Visual Perception

- Any design should have a colour scheme
    1. Monochromatic - Different tones of the same colour
    2. Analogous - Colours adjacent to each other on the colour wheel
    3. Complementary - Colours opposite each other on the colour wheel
    4. Triadic - Colours evenly spaced across the colour wheel
- Colours have cultural meaning

2.3 Types of Device

- Multimedia Input - Cameras, Microphones, Motion Sensors, etc.
- Visual Display Devices
- Multimedia Output Devices - 3D printers, Interactive Tables

2.3.1 Interaction Spaces and Devices

- Small information devices
- Information environments
- Immerse virtual environments
- Multi-user work environments
- Speech/ motion/ gesture/ gaze-based input

2.3.2 Interfaces and Interaction

- Interfaces are defined by the IO devices used
- Defined by the user-experience supported by the interface design
    - Instructing - command based
    - Conversing - dialogue based
    - Manipulating - static interaction with the environment
    - Exploring - dynamic interaction with the environment
- Multimodal interfaces use two or more types of communication
    - Can be used to disambiguate

2.4 Affective Design

- Deals with how the user feels
- Two aspects of affective design
  - Appearance and style - The look and feel of the interface
  - Usability - How frustrating or pleasing the system is to use
- Anthropomorphism - making computers humanlike, in appearance and emotionally

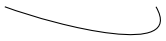
2.5 Task Analysis

- Process
  1. Identify the task to be completed
  2. Break this into subtasks
  3. Draw the subtasks as a layered diagram ensuring that it is complete
  4. Decide upon the level of detail into which to decompose
  5. Continually decompose the task
- Decompose into a tree, use arrows to show flow
- Symbols

Hierarchy

Sequence

Optional


Optional

3 Evaluation

3.1 Heuristic Evaluation

- Heuristic Evaluation - As if the evaluator were using it for the first time
- Evaluate the product against the list of 10 heuristics
  1. Know what is going on
  2. Follow real world conventions
  3. Make the user feel safe - home, always know you can get home(screen)
  4. Consistency between applications

5. Stop errors and tell them what they have done wrong in the case there is an error
6. Minimise user's memory load
7. Accelerators for experts
8. Draw user's attention to focus on main subject at hand
9. Should have the structure of the aw snap page, help recover from errors
10. Good documentation

- Rate the severity of the problem - How urgent the problem is, whether it has to be redesigned

Rating	Description	Example
0	No problem	System without fault
1	Cosmetic problem	Font isn't great
2	Minor usability problem	May slow down the user's completion of the task
3	Major usability problem	May stop the user being able to complete the task
4	Usability catastrophe	User unable to complete task

3.2 Cognitive Walkthrough

- Looking at how the user uses product first time
- Done to
  - Question assumptions about what users are thinking
  - Find controls that are missing or hard to find
  - Find places with inadequate feedback
- Has the following stages
  1. Define inputs - identify users, tasks, and actions to achieve tasks
  2. Get Analysts - typical developer can perform cognitive walkthrough, imagine a class of users
  3. Step through actions - Will users see/ know what to do
  4. Recording information - Record users knowledge before and after actions, assumptions about users
  5. Revise UI - Based on things identified in walkthrough fix
- Will find severe problems, content-related problems, finds specific problems
- But can not find how tasks interact